

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

**Absolvování individuální odborné
praxe
Individual Professional Practice in
the Company**

2010/2011

Pavel Čech

Zadání bakalářské práce

Student:

Pavel Čech

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

**Absolvování individuální odborné praxe
Individual Professional Practice in the Company**

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: XEVOS Solutions, s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Doc. Mgr. Jiří Dvorský, Ph.D.**

Konzultant bakalářské práce: Ing. Jan Pindřák

Datum zadání: 19.11.2010

Datum odevzdání: 06.05.2011



doc. Dr. Ing. Eduard Sojka
vedoucí katedry



prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Příboře 4.5.2011

.....

ABSTRAKT

Tato bakalářská práce popisuje průběh mé praxe, kterou jsem vykonával ve firmě XEVOS solution s.r.o. V úvodní části tuto společnost krátce představuji a zmiňuji se o náplni její práce. V dalších kapitolách rozebírám jednotlivé úkoly, které mi byly průběžně přidělovány. Podstatnou část praxe jsem se zabýval testováním, laděním a opravováním již implementovaných funkcí portálu. V závěru hodnotím přínos absolvované praxe pro můj další profesní rozvoj.

Klíčová slova: xevos.cz, poptávkový portál, webová aplikace, ASP.NET, C#, LINQ, platební brána, CSS, HTML, email, Coolpoptavka.cz

ABSTRACT

This bachelor thesis describes the course of my practice, which I performed in the company XEVOS Solution s.r.o. In the introductory part I briefly introduce this company and I mention the content of her work. In subsequent chapters, I discuss the various tasks that I was assigned to keep. A substantial part of the practice, I dealt testing, debugging and correcting the portal features that have already been implemented. Finally, I evaluate the contribution of completed practice for my continuing professional development.

Keywords: xevos.cz, demands site, web application, ASP.NET, C#, LINQ, payment gateway, CSS, HTML, email, Coolpoptavka.cz

Seznam použitých symbolů a zkratek

ASP.NET	- Active Server Pages .NET
BI	- Business Intelligence
CLR	- Common Language Runtime
LINQ	- Language INtegrated Query
SQL	- Structured Query Language
URL	- Unique Resource Locator
SSL	- Secure Sockets Layer
SMTP	- Simple Mail Transfer Protocol
HTTP	- HyperText Transfer Protocol
TFS	- Team Foundation Server

Obsah

1	Úvod	5
2	Profil firmy a pracovní zařazení studenta	6
3	Popis a řešení zadaných úkolů	7
3.1	Studování dokumentace a prohlídka kódu	7
3.2	Průzkum konkurence	7
3.3	Platba bránou Paypal	7
3.4	Nabídky dodavatelů	9
3.5	Zobrazování kategorií poptávek a nabídek	10
3.6	Validace formulářů zadávání poptávek	11
3.7	Filtry pro zobrazení firem	11
3.8	Odesílání informačních emailů	12
3.9	Generátor emailů s přehledem nových poptávek	12
3.10	Hezké URL	14
3.11	Detail a prezentace firmy	14
4	Přibližný časový plán zadaných úkolů	16
5	Scházející znalosti a získané dovednosti v průběhu odborné praxe	18
6	Závěr	19
7	Reference	20

Seznam tabulek

Tabulka 1: Září.....	16
Tabulka 2: Říjen.....	16
Tabulka 3: Listopad.....	16
Tabulka 4: Prosinec.....	16
Tabulka 5: Leden.....	16
Tabulka 6: Únor.....	17
Tabulka 7: Březen.....	17

Seznam obrázků

Obrázek č. 2: Pohled na přehled kategorií a přepínač poptávek a nabídek	10
Obrázek č. 4: Ukázka vygenerovaného emailu	13
Obrázek č. 5: Příklad hezké URL adresy	14

Seznam výpisu zdrojového kódu

Výpis č. 1: Konfigurace SMTP klienta pro testování odesílání emailu přes Gmail	12
Výpis č. 2: Ukázka nastavování parametrů emailu	12
Výpis č. 3: Ukázka nastavení v souboru web.config	14

1 Úvod

Bakalářská práce se zabývá průběhem mé padesátidenní praxe absolvované ve firmě XEVOS solution, s.r.o. Ve firmě jsem se zabýval hlavně programováním v C#, LINQ a ASP.NET. Podstatnou část času mi zabralo testování, ladění a opravování již hotových funkcí na internetových stránkách. Po celou dobu praxe jsem pracoval na řešení úkolů týkajících se vývoje poptávkového portálu coolpoptavky.cz.

V bakalářské práci popisuji profil firmy, přehled obsáhlejších zadáných úkolů a jejich následné řešení. Dále uvádím přibližný časový plán zadáných úkolů. V závěru shrnuji získané zkušenosti a celkové hodnocení praxe.

Po celou dobu praxe jsem na projektu úzce spolupracoval s dalším studentem Jakubem Fojtů. Na řešení většiny úkolů jsme se podíleli společně a pro jejich zpracování do bakalářské práce jsme si je proto rovnoměrně rozdělili.

2 Profil firmy a pracovní zařazení studenta

XEVOS Solutions poskytuje od svého založení v roce 2006 komplexní služby v oblasti informačních a komunikačních technologií s důrazem na inovativnost a vysokou dostupnost realizovaných řešení. Společnost se zaměřuje především na konzultace, realizace a správu serverů, síťové infrastruktury a klientských stanic v segmentu firem působících na území celé republiky.

V roce 2008 se specializace firmy rozšířila o vývoj webových aplikací na platformě ASP.NET, kde mezi pilotní projekty patří vývoj webového ERP systému s přidanou hodnotou postaveného čistě jako webová aplikace. Tato expanze vyústila v roce 2008 založením společnosti XEVOS Solutions s.r.o. a posílením týmu o nové členy. Dnes se orientuje také na Business Intelligence a inovaci stávajících aplikací o funkce využívající BI. [1]

Ve firmě jsem pracoval jako programátor s jazyky C# a ASP.NET. C# je vysokoúrovňový objektově orientovaný programovací jazyk, který vyvinula firma Microsoft. ASP.NET je technologie pro tvorbu webových aplikací. Je součástí .NET Frameworku a je založena na CLR. Projekty vytvářené za pomoci ASP.NET se tak můžou realizovat v jakémkoliv jazyce podporujícím CLR. [2]

Pro přístup k datům jsem ve firmě využíval jazyk LINQ. Tento jazyk dovoluje provádět operace nad daty jednoduchým způsobem pomocí dotazů podobných jazyku SQL.

Firma využívá pro vývoj webových aplikací produkt Telerik RadControls for ASP.NET AJAX, což jsou ovládací prvky webu od firmy Telerik. Bylo tedy nutné si pro práci ve firmě nainstalovat knihovny tohoto produktu a naučit se s nimi dále pracovat.

3 Popis a řešení zadaných úkolů

V průběhu praxe mi byly přidělovány úkoly týkající se tvorby nových funkcí a dále opravy, ladění a testování již implementovaných funkcí na poptávkovém portálu coolpoptavky.cz. Na portále si uživatelé zadávají poptávky, na které mohou registrované dodavatelské firmy reagovat formou své nabídky. Dodavatelé rovněž mohou na web vkládat své veřejné nabídky. Každá registrovaná dodavatelská firma je zařazena do katalogu, ve kterém si vytvoří svou vlastní prezentaci. V pravidelných intervalech je firmám zaslán seznam aktuálních poptávek, které by je mohly zajímat.

V následující části podrobněji rozeberu rozsáhlejší úkoly, na kterých jsem pracoval.

3.1 Studování dokumentace a prohlídka kódu

Pro efektivitu práce bylo nutné si nejprve nastudovat dokumentaci a zdrojové kódy projektu. Projekt coolpoptavky.cz vychází z diplomové práce ing. Jana Pindáka a v době, kdy jsem na něm začal pracovat, byl již v pokročilém stadiu rozpracování. Jelikož byl projekt pravděpodobně několikrát přepracováván, některé jeho funkce nebyly zcela zdokumentovány. To bylo pro mě docela velkým problémem, který mě provázel po celou dobu praxe. Trvalo dlouho, než mi bylo jasné, co jaký kus kódu dělá a jak ho ovlivním svým zásahem.

3.2 Průzkum konkurence

Cílem bylo prohledat internet a najít co nejvíce konkurenčních poptávkových portálů, následně je prozkoumat a zdokumentovat, jaký rozsah služeb zákazníkům poskytují a také co by se dalo použít jako inspirace pro firemní projekt.

Našel jsem okolo dvaceti podobných webů. Mezi významné funkce, které jsem našel a následně byly do portálu zapracovány, bylo například zadávání nabídek firem nezávisle na poptávkách. Doposud totiž mohly dodavatelské firmy pouze reagovat svou nabídkou na již existující poptávku na portále.

3.3 Platba bránou Paypal

Aby se dodavatelům mohly zobrazovat poptávky zadané na serveru, musí si nejprve objednat a zaplatit některý z nabízených tarifů. Z tohoto důvodu bylo nutné do portálu integrovat platební brány. Byl mi tedy přidělen úkol implementovat platbu přes systém Paypal.

Systém Paypal poskytuje celou řadu způsobů jak provést platbu. Standardní platby umožňuje provádět prostřednictvím odkazu nebo tlačítka umístěného na stránkách a směřujícího na stránky Paypal. Jedná se například o tlačítka „Donate“ či „Buy now“.

Pro účely portálu jsem měl ale využít takzvané expresní zaplacení využívající Paypal API. To zjednodušuje proces platby a kupující po zaplacení stále zůstává na stránkách portálu.

Pokud se dodavatel rozhodne uhradit platbu za tarif pomocí této platební brány, se ještě předtím, než je uživatel přesměrován na Paypal bránu, odešle HTTP požadavek a společně s ním se pošle metodou POST několik údajů o platbě. Na základě těchto dat platební brána vygeneruje a vrátí Token, který jednoznačně identifikuje platbu. Následně je sestavena

URL která obsahuje vygenerovaný Token a na ní je platící uživatel přesměrován. Zde se přihlásí na svůj Paypal účet, kde se mu zobrazí informace o tom, kam transakce směřuje a po kliknutí na tlačítko „continue“ je přesměrován zpět na náš web. Společně s tímto přesměrováním pošle brána i několik údajů metodou GET. Nejdůležitější z nich jsou Token a PayerId. Ty vygeneruje Paypal brána a jsou potřebné pro následné potvrzení platby, které se provede po kliknutí na tlačítko „potvrdit“ na našem webu. Token má platnost pouze 15 minut, do této doby musí být platba potvrzena.

Obrázek č. 1: Ukázka zobrazení informací o tom, kam platba směřuje.

3.3.1 Popis implementace platební brány

V místě, kde se platba inicializuje, jsem vytvořil metodu, která se stará o vytvoření a vygenerování údajů pro platební bránu. Automaticky si načte veškeré informace k platbě. V parametrech potřebuje pouze předat cenu a číslo platby. Rovněž tato metoda stará o získání Tokenu od platební brány a sestavuje URL, na kterou je poté platící uživatel přesměrován a kde se přihlašuje na svůj Paypal účet.

Dále jsem vytvořil třídu „Platba_Paypal_review“, která se stará o přijetí údajů z brány a obstarává následné potvrzení platby. Také tato třída ukládá výsledky transakce do databáze.

Pro sestavování HTTP požadavku pro získání Tokenu jsem použil pomocnou třídu NVPAPICaller, kterou dává k dispozici Paypal na svém webu. [3] Metody z této třídy jsem rovněž použil i pro odesílání potvrzení o platbě.

3.4 Nabídky dodavatelů

Aby dodavatelská firma na portále nemusela být aktivní pouze při posílání nabídek na již existující poptávky, měla by mít možnost své služby prezentovat formou samostatných nabídek na portále.

3.4.1 Vkládání nabídek

Měl jsem za úkol vytvořit formulář pro vkládání nabídky. Na formulář se dá dostat pouze z odkazu, který jsem přidal do menu účtu dodavatele.

Do formuláře jdou zadat následující informace:

- Předmět nabídky
- Popis nabídky
- Kategorie, do které nabídka spadá
- Platnost nabídky

Informace o zadavateli se do nabídky vloží automaticky po odeslání formuláře.

Vkládání nabídky zajišťuje jediný webový formulář, který jsem nazval „Nova_nabidka.aspx“. Pro zadání popisu a předmětu nabídky jsem použil komponentu TextBox. Pro usnadnění vkládání data platnosti nabídky jsem do formuláře vložil telerik komponentu RadDataPicker, která zobrazí malý kalendář a na něm lze datum pohodlně zadat. Pro zadání kategorií jsem vytvořil trojici rozbalovacích seznamů. Na korektní vkládání údajů uživatelem jsem vyrobil sadu validátorů.

Po odeslání formuláře se data uloží do databáze a přidá se k nim ID dodavatele jako ukazatel na zadavatele nabídky.

3.4.2 Zobrazování nabídek

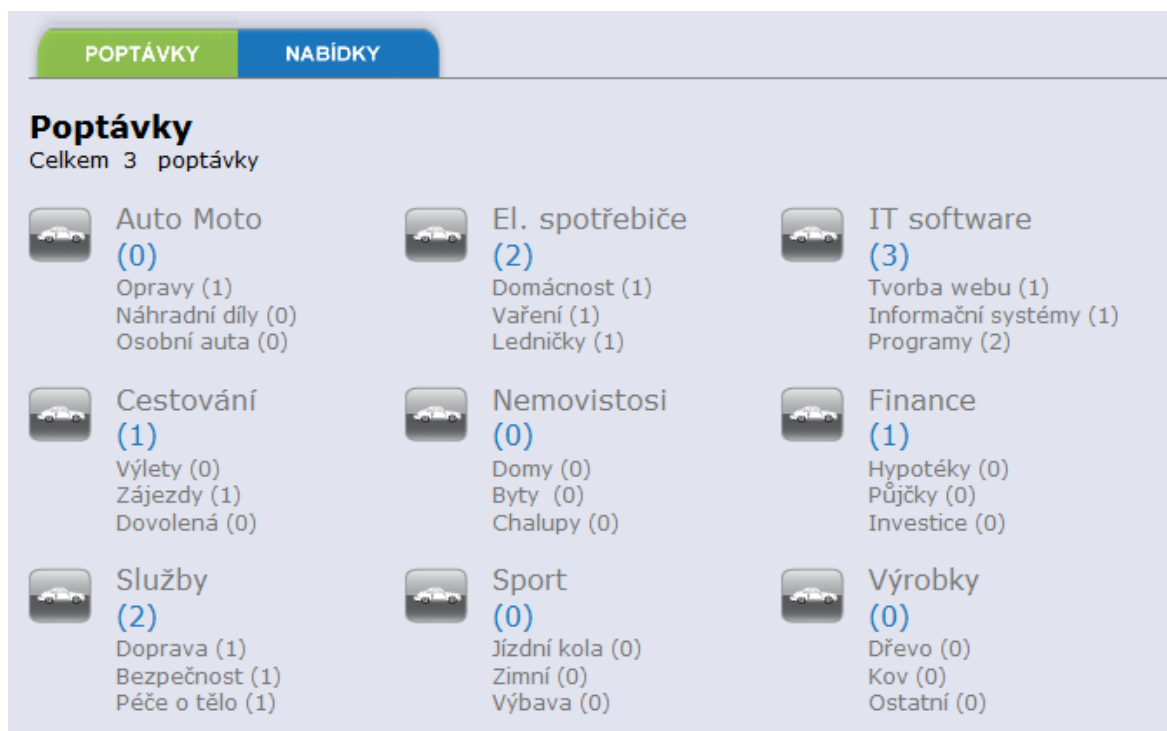
Detail veřejné nabídky dodavatelů si může prohlédnout jakýkoliv návštěvník webu. Každá nabídka je zobrazována pouze po dobu své platnosti. Pokud je účet firmy z nějakého důvodu pozastaven, jednotlivé nabídky se nezobrazují.

Pro zobrazení přehledu nabídek jsem vytvořil nový webový formulář. Jelikož je strukturou hodně podobný formuláři s přehledem poptávek, který byl již implementován, nemusel jsem jej pracně vymýšlet. Pouze jsem poměnil LINQ dotazy, aby vytahovaly z databáze platné nabídky.

Pro formulář s detailem nabídky jsem se rovněž dal použít kód z formuláře detailu poptávky. Musel jsem ale odstranit omezení, které u poptávky bylo a které zabraňovalo zobrazení kontaktu nepřihlášeným uživatelům.

3.5 Zobrazování kategorií poptávek a nabídek

Při zadávání poptávky na portále musí poptávající pro přehlednost vždy vybrat i kategorii a podkategorie, do kterých poptávka spadá. Tyto kategorie jsou pak viditelné přímo z hlavní stránky portálu. Mým úkolem bylo upravit funkčnost kategorií tak, aby se správně zobrazovaly na úvodní stránce. Také jsem měl vymyslet, jak se budou přepínat na úvodní stránce kategorie nabídek a poptávek.



Obrázek č. 2: Pohled na přehled kategorií a přepínač poptávek a nabídek

Začal jsem výrobou přepínacího tlačítka. Pro jeho výrobu jsem použil opensource grafický rastrový editor Gimp.

Celé okno s kategoriemi poptávek jsem vložil do komponenty Panel. To samé jsem udělal i pro okno s nabídkami, akorát jsem je nastavil jako výchozí neviditelné. Následně jsem vytvořil 2 metody, které se vyvolají při kliknutí na přepínací tlačítko. Jedna nastavuje zobrazení poptávek a zneviditelnění nabídek a druhá funguje obráceně.

Okna s poptávkami a nabídkami jsem realizoval pomocí komponent Repeater, kdy jsem z databáze vytáhl názvy kategorií a nastavil je jako datový zdroj pro Repeater. Komponenta Repeater je určena přímo pro zobrazování seznamů, její hlavní vlastností je plná kontrola nad výstupním HTML kódem.

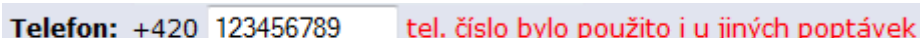
Formátování kategorií jsem provedl pomocí CSS a SeparatorTemplate, což je šablona pro oddělovač mezi položkami.

3.6 Validace formulářů zadávání poptávek

Popis poptávky na portále je ale viditelný pro všechny, avšak detail zadané poptávky a kontakty na poptávajícího jsou viditelné pouze ze zaplaceného účtu dodavatelských firem. Bylo tedy nutné zajistit, aby poptávající uživatel ve formuláři pro zadání poptávky nevyplnil v kolonce pro popis poptávky nějaký kontaktní údaj. Pro řešení tohoto problému jsem vytvořil validátor, který při rozporu znemožní odeslání poptávky. Tento validátor obsahoval regulární výraz, který vyhledával emailové adresy. Také jsem porovnával číslo zadané v kolonce pro telefonní kontakt, zda se nevyskytuje v poli pro popis poptávky. Pokud byl v popisu nalezen nějaký rozpor, validátor zamezil odeslání formuláře a zobrazil upozorňovací hlášku.

Další částí validace bylo zobrazení upozornění, pokud při zadávání telefonního kontaktu a emailu již tento kontakt byl použit. U požadované komponenty jsem nastavil parametr `AutoPostBack` na „true“ a poté jsem v metodě `Page_Load` na základě atributu `IsPostBack` vytvořil podmínku. Pokud je `IsPostBack` nastaven na „true“, zadaný kontakt se porovná s databází a podle případné shody zobrazí informační upozornění.

Pokud tedy uživatel vyplní obsah políčka pro telefon či email, vyvolá se `PostBack` a provedou se výše zmíněné kroky v metodě `Page_Load`.



Obrázek č. 3: Ukázka použití validátoru

3.7 Filtry pro zobrazení firem

V administraci v sekci s přehledem a editací registrovaných firem bylo potřeba zavést filtr pro přehled firem, kterým vypršel zaplacený tarif, nebo jim v nejbližší době vyprší.

Po delší úvaze jsem usoudil, že pro realizaci tohoto filtru je potřeba zavést v databázi do tabulky s účty firem nový atribut, který by obsahoval datum vypršení zaplaceného tarifu. Na základě tohoto data pak filtr pracoval.

Musel jsem naimplementovat metodu pro výpočet a ukládání tohoto data do databáze. Bylo potřeba k aktuálnímu datu přičíst délku využívání zaplaceného tarifu, která byla dostupná z databáze. Tento kód bylo nutné doplnit všude tam, kde se manipuluje s placením, či prodlužováním platnosti účtu.

Filtr jsem graficky realizoval sadou přepínacích komponent `RadioButton`, podle kterých bylo možné vyhledávat firmy, které mají zaplatit do určitého časového období.

Délku tohoto období jsem zvolil:

- 7 dnů
- 14 dnů
- 30 dnů
- Prošlá platba

Pokud uživatel přepínač nastaví, vyvolá se metoda obsahující LINQ dotaz, který na základě zvoleného přepínače vybere patřičné firmy. Tyto firmy pak zobrazím do tabulky.

3.8 Odesílání informačních emailů

Pro komunikaci portálu s uživateli je hojně využíváno emailových zpráv. Jsou zasílány nejrůznější druhy emailů, jedná se například o výzvy k zaplacení, faktury, informace o pozastavení účtu z nějakých konkrétních důvodů, přijetí nové zprávy na účet. Zasílána jsou také upozornění na novou nabídku pro zadanou poptávku.

Hrubý návrh odesílání emailu byl na portále již implementován, ale dlouhou dobu se mi nedařilo otestovat jeho funkčnost na mém lokálním serveru. Důvodem bylo blokování portu 25 poskytovatelem internetu v kanceláři, tento port slouží pro odesílání emailových zpráv. Musel jsem tedy přijít na způsob, jak tento problém obejít. V řešení mi byla nápomocna služba gmail od firmy Google. Ta používá k odesílání zpráv portu 587, který nebyl blokován. Avšak potřeboval jsem založit účet na gmail.com, v kódu je nutné použít autentifikaci a zapnout šifrování SSL. Pro odesílání emailu jsem použil třídu `Smtplib` a `MailMessage`.

```
//vytvoření instance smtpklienta
Smtplib client = new Smtplib("smtp.gmail.com", 587);
//zakázání výchozí autentifikace
client.UseDefaultCredentials = false;
//autentifikace
client.Credentials = new NetworkCredential("coolpoptavka@gmail.com", "xxx");
//zapnutí šifrování SSL
client.EnableSsl = true;
```

Výpis č. 1: Konfigurace SMTP klienta pro testování odesílání emailu přes Gmail

Pro potřebu odesílat více druhů informačních emailů jsem vytvořil třídu `Email_sender_controller`, která obsahovala sadu metod pro vytváření různých těl a hlaviček emailů. Poté se jen na místě pro odeslání upozornění zavolala patřičná metoda z této třídy pro nastavení parametru těla nebo hlavičky emailu.

```
//nastavení parametrů emailu
new Email().Send(toMail,
    Email_sender_controller.Send_suspension_subject(),
    Email_sender_controller.Send_suspension_body(duvod)
);
```

Výpis č. 2: Ukázka nastavování parametrů emailu

3.9 Generátor emailů s přehledem nových poptávek

Dodavatelské firmy, které mají na portále aktivní účet, si mohou nastavit přijímání informačních emailů se seznamem nových poptávek, které přibýly do kategorií, ve kterých se dodavatelská firma nachází. Dodavateli se tak podstatně usnadní přehled o poptávkách na portále. Tento informační email se generuje a odesílá pravidelně každý den v určitý čas.

Mým úkolem bylo naprogramovat funkci, která tento email vytvoří a odešle. Poté se tato funkce vloží do naplánovaných úloh. Email měl být v HTML formátu, dále měl obsahovat hlavičku a patičku a měl být po grafické stránce pěkný na pohled.

3.9.1 Popis implementace generátoru

Generátor fungoval formou cyklu, ve kterém jsem vybíral jednotlivou dodavatelskou firmu a u ní jsem následně zjistil, do jakých kategorií spadá. Poté jsem v cyklu procházel poptávky, které přibýly za poslední období a porovnával jejich kategorie s kategoriemi firmy. Pokud se kategorie shodovaly, přidal jsem poptávku do seznamu. Následně jsem předal seznam a ID firmy metodě, která z nich vytvořila obsah emailové zprávy a pomocí ID firmy odeslala email na správnou adresu. Toto celé se opakovalo tak dlouho, dokud se cyklus s firmami neukončil.

Pro odesílání emailů jsem použil třídy Smtplib a MailMessage, přičemž bylo nutné nastavit u instance zprávy parametr IsBodyHtml na „true“, aby tělo emailu bylo v HTML formátu.

Takto navržený generátor ale nefungoval úplně správně, protože pro kategorie dodavatelských firem a poptávek byly vytvořeny dvě na sobě nezávislé tabulky. Nebylo tedy možné porovnávat kategorie. Proto jsem sloučil tyto dvě tabulky do jedné. To si vyžádalo velké úpravy v administraci, kdy jsem musel kompletně předělat editaci kategorií. Také bylo potřeba předělat katalog firem a najít a upravit prakticky každý kód, u kterého se používala data z výše uvedených tabulek.

Hodně velkým problémem bylo vytvořit email tak, aby se při jeho zužování a rozšiřování grafická struktura příliš nenarušila. Toho jsem docílil pomocí CSS pozicování, kdy jsem u jednotlivých bloků použil procentuální velikost.

Přehled aktuálních poptávek

Pro dodavatele: **Personality s.r.l.**



Poptávám zimní pneu

Datum zadání: 1.3.2011 12:38:08
Kategorie: [Auto Moto](#)
Kraj: Moravskoslezský

Popis:
 Koupím ojeté zimní pneumatiky ve větším počtu.

[\[Detail poptávky\]](#)

Kontakt na poptávajícího:

Kontakt na osobu
Jméno: Chuan Dong
Email: [chuan@dong.cz](#)
Telefon: 703840088
Ulice: Koutek 2
Město, PSČ: Olomouc 77900

Poptávám vyřazené velké pneumatiky

Datum zadání: 1.3.2011 15:41:18
Kategorie: [Auto Moto](#)
Kraj: Jihočeský

Popis:
 Pro účely hrazení na motokárovou dráhu.

[\[Detail poptávky\]](#)

Kontakt na poptávajícího:

Kontakt na osobu
Jméno: Pavel Gernik
Email: [pavel@gernik.cz](#)
Telefon: 733661363
Ulice: Olomouc 20
Město, PSČ: Olomouc 77900

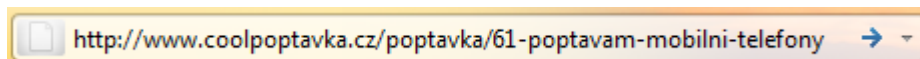
Tipy:
 - Změnit nastavení zasílání poptávek můžete udělat na svém účtu na [coolpoptavka.cz](#)
 - Vlastní poptávku můžete zadat [ZDE](#) - V případě potřeby nás prosím kontaktujte na e-mail [info@coolpoptavka.cz](#) nebo na telefonu +420 111 222 333

Přejeme Vám mnoho úspěchů při získávání nových zakázek.
 Tým [coolpoptavka.cz](#).

Obrázek č. 4: Ukázka vygenerovaného emailu

3.10 Hezké URL

Pojmem hezké URL můžeme označit odkazy na umístění stránek webu, které již v samotném odkazu obsahují informaci o obsahu stránky. Toto je jeden z faktorů pro dobré zaindexování webu do vyhledávačů.



Obrázek č. 5: Příklad hezké URL adresy

Hezké URL adresy byly již přibližně na půlce portálu funkční. Dostal jsem za úkol dodělat zbývající část.

Dosavadní hezké URL byly realizovány pomocí vlastních `HttpHandler`u. Držel jsem se tedy tohoto schématu.

`HttpHandler` je třída, která mapuje konkrétní HTTP požadavek na určitý koncový bod, například soubor, který odpovídá danému požadavku.

Nejprve jsem si musel vytvořit třídu (`HttpHandler`) implementující rozhraní `IHttpHandlerFactory`, která zpracovává předané URL a implementovat metodu `GetHandler`. Dále jsem v souboru `web.config` v sekci `system.web` nastavil filtr, aby výše vytvořený `HttpHandler` věděl, jaké požadavky má zpracovat.

Toto jsem udělal pro každou část portálu, kde byly hezké URL žádoucí.

```
<httpHandlers>
    ...
    <add verb="*" path="Poptavka/*.aspx" type="RemapUrl_poptavka" />
    ...
</httpHandlers>
```

Výpis č. 3: Ukázka nastavení v souboru `web.config`

3.11 Detail a prezentace firmy

Každá dodavatelská firma má na portále svůj profil, ze kterého se může poptávající uživatel o firmě dozvědět více informací.

Firma si může do svého profilu zadat tyto informace:

- Popis firmy do velikosti 2000 znaků
- Firemní logo
- Kontaktní osobu
- Kontaktní emaily a telefony
- Odkaz na webové stránky
- GPS souřadnice
- Provozní dobu
- Galerii fotografií

Adresa firmy a IČO se u profilu firmy zobrazují automaticky.

V případě, že firma vlastní nějakou pobočku, může si ji do profilu také přidat. Nastavit u ní lze zobrazování stejných informací jako u firmy.

Z menší části již byla stránka zobrazující profil firmy hotová, ale údaje na ní byly zatím jen testovací a byly zadány staticky. Mým úkolem bylo tuto stránku upravit do konečné podoby a graficky doladit.

Pro zobrazování jednotlivých firemních údajů jsem do souboru aspx s uživatelským rozhraním vytvořil sadu panelů a u nich nastavil atribut viditelnosti na „false“. V každém panelu se nacházel blok pro zadávání informací, a po jeho vyplnění se zviditelnil.

Nejobtížnější bylo vymyslet rozestavení bloků tak, aby se jejich různou viditelností nerozbil design stránky. Toho jsem docílil díky kombinaci kaskádových stylů a HTML tabulky.

4 Přibližný časový plán zadaných úkolů

Měsíc září

-
- Seznamování se se specifikacemi a dokumentací projektu coolpoptavka.cz.
 - Instalace potřebných aplikací pro práci na projektu.
 - Prohlídka kódu projektu coolpoptavka.cz.
-

Počet odpracovaných hodin: cca 55hod

Tabulka 1: Září

Měsíc říjen

-
- Průzkum konkurenčních poptávkových webů, hledání a sepisování nových funkcí.
 - Testování funkcí portálů a sepisování chyb.
 - Studování a učení jazyka LINQ.
-

Počet odpracovaných hodin: cca 53hod

Tabulka 2: Říjen

Měsíc listopad

-
- Implementace platby přes Paypal.
 - Implementace podkategorií dodavatelských firem.
-

Počet odpracovaných hodin: cca 56 hod

Tabulka 3: Listopad

Měsíc prosinec

-
- Implementace podkategorií poptávek.
 - Analýza úprav portálu na vkládání nabídek.
 - Implementace zadávání a zobrazování nabídek.
-

Počet odpracovaných hodin: cca 37 hod

Tabulka 4: Prosinec

Měsíc leden

-
- Ladění a testování procesů s nabídkami.
 - Validátor formuláře zadávání poptávek.
-

Počet odpracovaných hodin: cca 30 hod

Tabulka 5: Leden

Měsíc únor

- Implementace filtru dodavatelských firem v administraci.
 - Implementace odesílání různých upozornění a informačních zpráv na email.
 - Generování a odesílání emailů s přehledem nových poptávek.
-

Počet odpracovaných hodin: cca 91 hod

Tabulka 6: Únor

Měsíc březen

- Implementace odesílání upozorňovacího emailu zadavateli poptávky bez aktivního účtu při reakci dodavatele na jeho poptávku.
 - Dodělání hezkých URL.
 - Implementace profilu firmy a implementace zadávání údajů do profilu.
 - Implementace platby MojePlatba od komerční banky (nebyla dokončena, protože nebyl do konce praxe dodán certifikát k testování platby).
 - Zadávání testovacích záznamů a testování.
 - Sepisování, co vše se ještě musí na portále dodělat před jeho nasazením.
-

Počet odpracovaných hodin: cca 80 hod

Tabulka 7: Březen

5 Scházející znalosti a získané dovednosti v průběhu odborné praxe

Ze začátku praxe jsem se potýkal s malou znalostí technologie ASP.NET, ale díky vstřícnosti ostatních kolegů, se kterými jsem sdílel kancelář, se mi dařilo úspěšně plnit zadané úkoly a postupem času jsem se osamostatnil.

Poprvé jsem si vyzkoušel co je to teamově programovat a naučil jsem se pracovat s platformou TFS, která umožňuje efektivní spolupráci na vývoji software.

Také jsem se poprvé setkal s jazykem LINQ, ale díky databázovým předmětům a znalosti SQL ze školy, mi jeho pochopení a naučení nedělalo příliš velký problém.

6 Závěr

Projekt coolpoptavka.cz se za dobu mé praxe posunul vpřed a obohatil jsem jej o řadu nových funkcí. Přidělené úkoly jsem se snažil splnit co nejlépe v rámci svých možností a schopností. U většiny jsem dosáhl požadovaných výsledků, ale jistě by bylo mnohé co zlepšit.

Praxe pro mě byla velkým přínosem. Dověděl jsem se a naučil spoustu nových věcí a získal cenné zkušenosti, které jistě v budoucnosti využiji. Jsem rád, že mi škola poskytla tvorbu bakalářské praxe touto formou.

7 Reference

- [1] *XEVOS Solutions* [online]. 2009 [cit. 2011-04-25]. Profil firmy. Dostupné z WWW: <<http://www.xevos.cz/?stranka=20-firemni-profil>>.
- [2] *MSDN* [online]. 2011 [cit. 2011-04-25]. ASP.NET Web Applications. Dostupné z WWW: <<http://msdn.microsoft.com/en-us/library/ms644563.aspx>>.
- [3] *PayPal Integration Wizards*: [online]. 2008 [cit. 2011-04-25]. Dostupné z WWW: <<https://www.paypal-labs.com/integrationwizard/ecpaypal/code.php>>.